Quadratic time algorithm for inversion of binary permutation polynomials

Lucas Barthelemy, Delaram Kahrobaei, Guénaël Renault, Zoran Šunić

Quadratic time algorithm for inversion of binary permutation polynomials*

Lucas Barthelemy[1], Delaram Kahrobaei[2][0000−0001−5467−7832], Guénaël Renault[3][0000−0002−7050−9975], and Zoran Šunić[4][0000−0001−7861−158X]

[1] Quarkslab, Paris, FRANCE lbarthelemy@quarkslab.com
[2] Graduate Center, CUNY, New York, NY, USA dkahrobaei@gc.cuny.edu
[3] Agence Nationale de la Sécurité des Systèmes d'Information, 51 boulevard de La Tour-Maubourg, 75700 Paris 07 SP, France
Sorbonne Université, UPMC, LIP6, 4 place Jussieu, F-75252 Paris cedex 5, France
guenael.renault@ssi.gouv.fr
[4] Department of Mathematics, Hofstra University, Hempstead, NY 11549 USA, zoran.sunic@hofstra.edu

Abstract. In this paper, we propose a new version of the Lagrangian interpolation applied to binary permutation polynomials and, more generally, permutation polynomials over prime power modular rings. We discuss its application to obfuscation and reverse engineering.

Keywords: Permutation polynomial · Lagrangian interpolation · Obfuscation.

1 Motivation and Introduction

Permutation polynomials in the context of Galois' fields are very well studied in particular for their applications in cryptography. The study of binary polynomials (polynomials with coefficients in an integer ring modulo a power of 2) is less extensive, but it has been shown recently that they are important for computer security. As discussed in [8] and in [1], a straightforward application of binary permutation polynomials is obfuscation. Here we define obfuscation as a way to write computer programs that prevents reverse engineering of applications while minimizing the overhead in memory/computation cost.

In comparison with finite field permutation polynomials, the binary polynomials allow fast computation of bijective functions, since they can be directly implemented with low level arithmetic operations on computers. Moreover their use adds diversity to obfuscation techniques. The last point is of primary concern for obfuscation. Indeed, obfuscation usually does not rely on one overwhelming method, but on an aggregation of several layers of different techniques that aim to prevent automated attacks. For example, in [1] new classes of polynomials were considered and proved to be resistant to the attacks defined in [2].

* The second-named author was partially supported by a PSC-CUNY grant from the CUNY Research Foundation and by the ONR (Office of Naval Research) grant N000141512164

In this application context, the study of permutation polynomials is purely algorithmic and a central operation is the computation of the inverse of such a polynomial.

Newton's method for inverting binary permutation polynomials is an effective algorithm, but we present in this paper a new technique based on Lagrangian interpolation with two important properties:

- In a designer point of view, it is very important to measure the strength of any obfuscation technique based on binary permutation polynomials. The interpolation algorithm analyzed in this paper is proven to have a fixed complexity. This provides a more precise framework when measuring attack complexities with regard to computational overhead of using a binary permutation polynomial.
- From the reverse engineering point of view, this algorithm enables inversion techniques in a black-box context (i.e. when an encoding function is given as an evaluation function only). This is of importance when considering the reliance of encodings based on binary permutation polynomials since this algorithm can retrieve the explicit function through interpolation.

In addition, our version of Lagrangian interpolation allows a better understanding on how to use binary permutation polynomials. This should prove useful for future work on the subject.

2 Interpolation of the inverse polynomial over $\mathbb{Z}_{2^n}$

2.1 Reduction of integer polynomials

Integer multiples of Newton polynomials may be used to reduce any integer polynomial to a polynomial of relatively small degree (no greater than $n+\log_2 n$) that induces the same function on $\mathbb{Z}_{2^n}$. The approach follows Mullen and Stevens [5] and has recently been used in [1] in the context of inversion of polynomials by using Largange interpolation and also Newton's method.

For $i \geq 0$, let t_i be the largest integer ℓ such that 2^ℓ divides $\ell!$, and let d_n be the largest integer i such that $n - t_i > 0$. Note that d_n is always odd and not greater than $n + \log_2 n$. Define

$$P_i(x) = 2^{n-t_i} \prod_{j=0}^{i-1}(x-j) \text{ for } i = 0, 1, \ldots, d_n, \text{ and } P_{d_n+1}(x) = \prod_{j=0}^{d_n}(x-j).$$

Each polynomial $P_i(x)$, for $i = 0, 1, \ldots, d_n + 1$, is an integer multiple of the Newton polynomial $\prod_{j=0}^{i-1}(x-j)$ of degree i, and only the last one, $P_{d_n+1}(x)$, is monic. The ideal I of $\mathbb{Z}[x]$ generated by $P_0(x), \ldots, P_{d_n+1}(x)$ consists precisely of all integer polynomials that induce the zero function on $\mathbb{Z}_{2^n}$. Define the set of reduced polynomials $\mathcal{R}_n$ as the set of all integer polynomials $b_0+b_1x+\cdots+b_{d_n}x^{d_n}$ of degree at most d_n, such that, for $i = 0, \ldots, d_n$,

$$0 \leq b_i < 2^{n-t_i}.$$

For every integer polynomial $P(x)$ there exists a unique reduced polynomial $P_R(x)$ such that $P(x)$ and $P_R(x)$ induce the same function on $\mathbb{Z}_{2^n}$. The reduction is performed as follows. First, $P(x)$ is replaced by its reminder modulo the monic polynomial $P_{d_n+1}(x)$, and this yields a polynomial $R(x)$ of degree at most d_n. If $R(x)$ is reduced we are done. Otherwise, let i be the largest degree such that the i-coefficient of $R(x)$ is not in the range from 0 to $2^{n-t_i}-1$ and let c_i be the value of this coefficient. Then there exists a nonzero q such that $c_i = 2^{n-t_i}q + r$, where $0 \leq r \leq 2^{n-t_i}-1$. Thus, the i-coefficient of the polynomial $R(x) - qP_i(x)$ is equal to r, which is in the correct range. Continuing in the same fashion we may push all coefficients, one by one, in the order from highest to lowest degree, into the correct range and obtain a reduced polynomial.

2.2 Precise description of the inversion problem

If the degree of the original integer polynomial $P(x)$ is high and/or if its coefficients are large integers, the reduction procedure may take a long time. We are not interested in this issue, our quadratic algorithm assumes that $P(x)$ is given either in reduced form or as a black box that can calculate the sequence of values $P(0), P(1), \ldots, P(d_n)$ in $\mathbb{Z}_{2^n}$ in $O(n^2)$ time. Note that if we are given a reduced polynomial $P(x)$, then we can calculate $P(0), P(1), \ldots, P(d_n)$ in $O(n^2)$ time.

We formulate precisely the input and output for our problem.

Let $P(x)$ be an integer polynomial that induces a permutation on $\mathbb{Z}_{2^n}$.

Input: the sequence of values $P(0), \ldots, P(d_n)$ in $\mathbb{Z}_{2^n}$.

Output: the sequence of coefficients $b_0, b_1, \ldots, b_{d_n}$ of the unique reduced polynomial $Q(x)$ that induces the inverse permutation to $P(x)$ on $\mathbb{Z}_{2^n}$.

We know with certainty that a polynomial solution exists, since $P(x)$ induces a permutation on a finite set, which implies that some iteration of $P(x)$, which is also a polynomial with integer coefficients, induces the inverse permutation.

Our quadratic "time" complexity actually refers to the number of multiplications and/or additions and/or inversions of units in $\mathbb{Z}_{2^n}$ necessary to calculate the sequence of coefficients of $Q(x)$. The numbers involved in these calculations have $O(n)$ digits, but each addition/multiplication/inversion is counted as being performed in unit time.

We state our main result.

Theorem 1. *Let $P(x) \in \mathbb{Z}[x]$ be a polynomial that induces a permutation on $\mathbb{Z}_{2^n}$, given by its sequence of values $P(0), \ldots, P(d_n)$ in $\mathbb{Z}_{2^n}$. There exists an algorithm of time complexity $O(n^2)$ that determines the sequence of coefficients $b_0, b_1, \ldots, b_{d_n}$ of the unique reduced polynomial $Q(x)$ that induces the inverse permutation to $P(x)$ on $\mathbb{Z}_{2^n}$.*

2.3 Binary permutation polynomials

There is a simple characterization of binary permutation polynomials in terms of the coefficients of the polynomial. Namely, a polynomial $P(x) = a_0 + \cdots +$

$a_m x^m \in \mathbb{Z}[x]$ induces a permutation on $\mathbb{Z}_{2^n}$ if and only if (i) a_1 is odd, (ii) the sum $a_3 + a_5 + a_7 \ldots$ is even, and (iii) the sum $a_2 + a_4 + a_6 + \ldots$ is even.

The criterion is stated and proved in this form by Rivest [7], but he points out that it also follows easily from the following more general criterion: $P(x)$ induces a permutation on $\mathbb{Z}_{p^n}$, where p is a prime and $n \geq 2$, if and only if (i) $P(x)$ induces a permutation on $\mathbb{Z}_p$ and (ii) $P'(a) \not\equiv 0 \pmod{p}$ for $a \in \mathbb{Z}$. The last criterion is stated in the work of Mullen and Stevens [5], who consider it a direct corollary of Theorem 123. in the book by Hardy and Wright [3].

The following corollary is crucial for our purposes.

Corollary 1. *Let $P(x) = a_0 + \cdots + a_m x^m \in \mathbb{Z}[x]$ induce a permutation on $\mathbb{Z}_{2^n}$. For all $a, b \in R$, with $a \neq b$, the Newton quotient $k_{a,b} = \frac{P(a)-P(b)}{a-b}$ is an odd integer.*

Proof. Indeed, $k_{a,b} = a_1 A_1 + a_2 A_2 + \cdots + a_m A_m$, where $A_1 = 1$ and, for $i = 2, 3, \ldots, m$, $A_i = a^{i-1} + a^{i-2}b + \cdots + ab^{i-2} + b^{i-1}$. If both a and b are even then, modulo 2, $k_{a,b} \equiv a_1$, if they have different parity then $k_{a,b} \equiv a_1 + a_2 + \cdots + a_m$, and if they are both odd, $k_{a,b} \equiv a_1 + a_3 + a_5 + \ldots$. Thus, $k_{a,b}$ is always odd.

2.4 Solving the associated linear system

Fix n, and to simplify notation, set $d = d_n$.

For $i = 0, \ldots, d$, set $x_i = P(i)$ and $y_i = Q(x_i) = i$. We need to solve, over $\mathbb{Z}_{2^n}$, the linear system of equations

$$V[x_0, x_1, \ldots, x_d](b_0, b_1, \ldots, b_d)^T = (y_0, y_1, \ldots, y_d)^T,$$

where $V = V[x_0, x_1, \ldots, x_d] = [v_{i,j}]_{(d+1)\times(d+1)}$ is the $(d+1) \times (d+1)$ Vandermonde matrix in which $v_{i,j} = x_i^j$.

We will use the following two results by Oruç and Phillips.

Theorem 2 (Oruç-Phillips 2000 [6]). *Let $x_0, x_1, \ldots, x_m$ be distinct. An explicit LDU decomposition of the Vandermonde matrix $V = V[x_0, x_1, \ldots, x_m]$ is given by $V = LDU$, where D is the diagonal matrix*

$$Diag(1,\ x_1 - x_0,\ (x_2 - x_1)(x_2 - x_0), \ldots,\ (x_m - x_{m-1})(x_m - x_{m-2}) \ldots (x_m - x_0)),$$

L is the lower triangular matrix $L = [\ell_{i,j}]$ given by

$$\ell_{i,j} = \prod_{t=0}^{j-1} \frac{x_i - x_{j-1-t}}{x_j - x_{j-1-t}}, \qquad 0 \leq j \leq i \leq m,$$

and U is the upper triangular matrix $U = [u_{i,j}]$ given by

$$u_{i,j} = \tau_{j-i}(x_0, \ldots, x_i) \qquad 0 \leq i \leq j \leq m,$$

with the understanding that empty products are equal to 1 (thus all diagonal entries in both L and U are equal to 1), and $\tau_r(x_0, \ldots, x_i)$ is the complete symmetric function evaluated at $x_0, \ldots, x_i$, that is,

$$\tau_r(x_0, \ldots, x_i) = \sum_{\lambda_0 + \lambda_1 + \cdots + \lambda_i = r} x_0^{\lambda_0} x_1^{\lambda_1} \ldots x_i^{\lambda_i}.$$

Example 1. For $m = 4$, we have

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & \frac{x_2-x_0}{x_1-x_0} & 1 & 0 & 0 \\ 1 & \frac{x_3-x_0}{x_1-x_0} & \frac{(x_3-x_1)(x_3-x_0)}{(x_2-x_1)(x_2-x_0)} & 1 & 0 \\ 1 & \frac{x_4-x_0}{x_1-x_0} & \frac{(x_4-x_1)(x_4-x_0)}{(x_2-x_1)(x_2-x_0)} & \frac{(x_4-x_2)(x_4-x_1)(x_4-x_0)}{(x_3-x_2)(x_3-x_1)(x_3-x_0)} & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & x_0^4 \\ 0 & 1 & x_0+x_1 & x_0^2+x_0x_1+x_1^2 & x_0^3+x_0^2x_1+x_0x_1^2+x_1^3 \\ 0 & 0 & 1 & x_0+x_1+x_2 & x_0^2+x_0x_1+x_1^2+x_0x_2+x_1x_2+x_2^2 \\ 0 & 0 & 0 & 1 & x_0+x_1+x_2+x_3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The entries of U can be obtained recursively, by $u_{0,j} = x_0^j$, $u_{i,i} = 1$, and

$$u_{i,j} = u_{i-1,j-1} + u_{i,j-1} \cdot x_i, \qquad \text{for } 1 \le i < j. \tag{1}$$

Theorem 3 (Oruç-Phillips 2000 [6]). *Let $x_0, x_1, \ldots, x_m$ be distinct. The matrix L from the explicit LDU decomposition of the Vandermonde matrix $V = V[x_0, x_1, \ldots, x_m]$ given in Theorem 2 decomposes as the product*

$$L = L^{(1)}L^{(2)} \ldots L^{(m)}$$

of subdiagonal $(m+1) \times (m+1)$ matrices $L^{(k)} = [\ell_{i,j}^{(k)}]$ with 1s on the diagonal and the subdiagonal entries given, for $j = 0, \ldots, m-1$, by

$$\ell_{j+1,j}^{(k)} = \begin{cases} 0, & 0 \le j < m-k, \\ \prod_{t=0}^{j-(m-k)-1} \dfrac{x_{j+1}-x_{j-t}}{x_j - x_{j-1-t}}, & m-k \le j \le m. \end{cases}$$

Example 2. For $m = 4$, the following table provides the subdiagonal entries:

j	$\ell_{j+1,j}^{(1)}$	$\ell_{j+1,j}^{(2)}$	$\ell_{j+1,j}^{(3)}$	$\ell_{j+1,j}^{(4)}$
0	0	0	0	1
1	0	0	1	$\frac{x_2-x_1}{x_1-x_0}$
2	0	1	$\frac{x_3-x_2}{x_2-x_1}$	$\frac{(x_3-x_2)(x_3-x_1)}{(x_2-x_1)(x_2-x_0)}$
3	1	$\frac{x_4-x_3}{x_3-x_2}$	$\frac{(x_4-x_3)(x_4-x_2)}{(x_3-x_2)(x_3-x_1)}$	$\frac{(x_4-x_3)(x_4-x_2)(x_4-x_1)}{(x_3-x_2)(x_3-x_1)(x_3-x_0)}$

The subdiagonal entries can be calculated recursively as follows. For fixed j and $k = m-j$, we have $\ell_{j+1,j}^{(k)} = 1$, and for $k > m-j+1$,

$$\ell_{j+1,j}^{(k)} = \ell_{j+1,j}^{(k-1)} \cdot \frac{x_{j+1} - x_{m-k+1}}{x_j - x_{m-k}}. \tag{2}$$

Going back to our situation, we see that the entries of U and D are integers and, as such, are well defined over $\mathbb{Z}_{2^n}$. The entries of L and $L^{(k)}$ are not necessarily integers, but they are still well defined over $\mathbb{Z}_{2^n}$.

Proposition 1. *Let $P(x) \in \mathbb{Z}[x]$ induce a permutation on $\mathbb{Z}_{2^n}$.*

(a) Each entry of L is has odd denominator in its simplest form.

(b) Each entry of $L^{(k)}$, for $k = 1, \ldots, d$ has odd numerator and odd denominator in its simplest form.

Proof. (a) By Corollary 1, we have, for $0 \le j \le i \le d$,

$$\ell_{i,j} = \prod_{t=0}^{j-1} \frac{x_i - x_{j-1-t}}{x_j - x_{j-1-t}} = \prod_{t=0}^{j-1} \frac{P(y_i) - P(y_{j-1-t})}{P(y_j) - P(y_{j-1-t})} = \prod_{t=0}^{j-1} \frac{(y_i - y_{j-1-t})k_{i,j-1-t}}{(y_j - y_{j-1-t})k_{j,j-1-t}}$$
$$= \prod_{t=0}^{j-1} \frac{(i-j+1+t)k_{i,j-1-t}}{(1+t)k_{j,j-1-t}} = \binom{i}{j} \prod_{t=0}^{j-1} \frac{k_{i,j-1-t}}{k_{j,j-1-t}},$$

where each $k_{*,*}$ is an odd integer.

(b) Let $d - k \le j \le d$ and set $s = j - (d-k) - 1$. By Corollary 1, we have,

$$\ell^{(k)}_{j+1,j} = \prod_{t=0}^{s} \frac{x_{j+1} - x_{j-t}}{x_j - x_{j-1-t}} = \prod_{t=0}^{s} \frac{P(y_{j+1}) - P(y_{j-t})}{P(y_j) - P(y_{j-1-t})} = \prod_{t=0}^{s} \frac{(y_{j+1} - y_{j-t})k_{j+1,j-t}}{(y_j - y_{j-1-t})k_{j,j-1-t}}$$
$$= \prod_{t=0}^{s} \frac{(1+t)k_{j+1,j-t}}{(1+t)k_{j,j-1-t}} = \prod_{t=0}^{s} \frac{k_{j+1,j-t}}{k_{j,j-1-t}},$$

where each $k_{*,*}$ is an odd integer.

We are ready to prove the main result.

Proof (Proof of Theorem 1). Recall that, in our situation, $m = d = d_n < n + \log_2 n$ and we are solving the system $LDU\mathbf{b} = \mathbf{y}$. The recursive formulas (1) and (2) show that the entries of U and the subdiagonal entries in all $L^{(k)}$, $k = 1, \ldots, d$, can be calculated in $O(n^2)$ steps. The diagonal entries of D can also be calculated recursively in $O(n^2)$ steps. The inverse of $L^{(k)}$ is obtained by simply changing the sign in all subdiagonal entries. Therefore, we can calculate $\mathbf{y}' = L^{-1}\mathbf{y} = {L^{(m)}}^{-1}{L^{(m-1)}}^{-1} \ldots {L^{(1)}}^{-1}\mathbf{y}$ in $O(n^2)$ steps.

We then solve the system $DU\mathbf{b} = \mathbf{y}'$ by backward substitution in $O(n^2)$ steps. Note that the i-entry of D has the form $2^{t_i} f_i$, where f_i is odd. Because of our constraints on the coefficients of reduced polynomials, we are seeking only for solutions for b_i in the range $0 \le b_i < 2^{n-t_i}$, and a solution exists and is unique in this range. More precisely, once $b_{i+1}, \ldots, b_d$ are substituted in, we need to solve for b_i from an equation of the form $2^{t_i} f_i b_i = g_i \pmod{2^n}$, for some odd f_i and some $g_i \in \mathbb{Z}_{2^n}$. We already know that a solution exists, so it must be that $g_i = 2^{t_i} g_i'$ for some $g_i' \in \mathbb{Z}_{2^n}$. After canceling the term 2^{t_i} we solve for b_i from $f_i b_i = g_i' \pmod{2^{n-t_i}}$ by inverting f_i, and thus produce the unique solution in the range $0 \le b_i < 2^{n-t_i}$.

2.5 Another solution

If we are not interested in producing the coefficients of the inverse polynomial $Q(x)$, but rather just in calculating the values of $Q(x)$ at various points, a slightly different algorithm exists and we outline it here.

Let $\mathcal{U}_n$ be the ring of units of the ring $\mathbb{Z}_{2^n}$. Without loss of generality we may assume that $P(x)$ separately permutes $\mathcal{U}_n$, the odds, and its complement, the evens (if it does not, we may replace $P(x)$ by $P(x)+1$).

The ideal I' of integer polynomials that induce the zero function on $\mathcal{U}_n$ is described in [4]. It is generated by $P_i(x) = 2^{n-i-t_i}\prod_{j=0}^{i-1}(x-(2j+1))$, for $i = 0, 1, \ldots, d'_n$, and $P_{d'_n+1}(x) = \prod_{j=0}^{d'_n}(x-(2j+1))$, where d'_n is the largest integer i such that $n-i-t_i > 0$. Every integer polynomial that permutes $\mathcal{U}_n$ has a unique representative modulo I', which is a polynomial of degree at most d'_n with the i-coefficient in the range from 0 to $2^{n-i-t_i}-1$. The maximum degree d'_n is approximately half of d_n. We may calculate, by using the same approach as above (the Vandermonde matrix will have dimension $(d'_n+1)\times(d'_n+1)$ and the interpolation is preformed for $x_i = P(2i+1)$, $i = 0, \ldots, d'_n$) the coefficients of the unique reduced polynomial $\overline{Q}(x)$ modulo I' that inverts the values of $P(x)$ on $\mathcal{U}_n$ (and not necessarily on its complement).

By a similar approach, the coefficients of another polynomial, $\overline{\overline{Q}}(x)$, of degree at most d'_n that inverts the values of $P(x)$ on the complement of $\mathcal{U}_n$ may be calculated. The two polynomials $\overline{Q}(x)$ and $\overline{\overline{Q}}(x)$ may then be used to calculate the values of $Q(x)$ (use the former for odd x and the latter for even). Since the degrees of $\overline{Q}(x)$ and $\overline{\overline{Q}}(x)$ are, in general, smaller than the degree of $Q(x)$, this approach may be faster if we need to calculate many values of $Q(x)$.

3 Interpolation of the inverse polynomial over $\mathbb{Z}_{p^n}$

Fix a prime p and $n > 1$.

We claim that the same inversion technique works equally well for permutation polynomials over the ring $\mathbb{Z}_{p^n}$.

The ideal of integer polynomials that induce the zero function on $\mathbb{Z}_{p^n}$ is generated by the polynomials

$$P_i(x) = p^{n-t_{p,i}}\prod_{j=0}^{i-1}(x-j) \text{ for } i = 0, 1, \ldots, d_{p,n}, \text{ and } P_{d_{p,n}+1}(x) = \prod_{j=0}^{d_{p,n}}(x-j).$$

where, for $i \geq 0$, $t_{p,i}$ is the largest integer ℓ such that p^ℓ divides $\ell!$, and $d_{p,n}$ is the the largest integer i such that $n - t_{p,i} > 0$. Each integer polynomial is equivalent, as a function over $\mathbb{Z}_{p^n}$, to a unique reduced polynomial, that is, polynomial $b_0 + b_1 x + \cdots + b_{d_{p,n}} x^{d_{p,n}}$ of degree at most $d_{p,n}$, such that, for $i = 0, \ldots, d_{p,n}$, we have $0 \leq b_i < p^{n-t_{p,i}}$ (see [5, Theorem 2.1]).

We prove an analog of Corollary 1.

Proposition 2. *Let $P(x) = a_0 + \cdots + a_m x^m \in \mathbb{Z}[x]$ induce a permutation on $\mathbb{Z}_{p^n}$. For all $a, b \in \mathbb{Z}$, with $a \neq b$, the Newton quotient $k_{a,b} = \frac{P(a)-P(b)}{a-b}$ is an integer that is not divisible by p.*

Proof. Since $P(x)$ induces a permutation on $\mathbb{Z}_{p^n}$, it induces a permutation on $\mathbb{Z}_p$ and $P'(a) \not\equiv 0 \pmod{p}$, for all a.

We work modulo p.

Let $a \neq b$ and, moreover, $a - b \not\equiv 0$. If we assume $k_{a,b} \equiv 0$, then $P(a) - P(b) \equiv (a-b)k_{a,b} \equiv 0$, a contradiction, since $P(x)$ permutes $\mathbb{Z}_p$.

Let $a \neq b$, but $a \equiv b$. Then, for $i \geq 2$, we have $A_i = a^{i-1} + a^{i-2}b + \cdots + ab^{i-2} + b^{i-1} \equiv ia^{i-1}$ and $k_{a,b} \equiv a_1 + 2a_2 a + \cdots + m a_m a^{m-1} \equiv P'(a) \not\equiv 0$.

The rest of the proof is exactly the same as in the binary case, except, of course, that the analog of Proposition 1 should state that, in their simplest form, all denominators of the entries in L are integers not divisible by p, and all numerators and denominators of the entries in $L^{(k)}$, $k = 1, \ldots, d_{p,n}$, are integers not divisible by p. Thus, L and $L^{(k)}$ are well defined over $\mathbb{Z}_{p^n}$.

Thus we may state a more general version of our main result.

Theorem 4. *Let p be a prime and $P(x) \in \mathbb{Z}[x]$ a polynomial that induces a permutation on $\mathbb{Z}_{p^n}$, given by its sequence of values $P(a), P(a+1), \ldots, P(a+d_{p,n})$ in $\mathbb{Z}_{p^n}$ for some $a \in \mathbb{Z}_{p^n}$ (not necessarily 0). There exists an algorithm of time complexity $O(n^2)$ that determines the sequence of coefficients $b_0, b_1, \ldots, b_{d_{p,n}}$ of the unique reduced polynomial $Q(x)$ that induces the inverse permutation to $P(x)$ on $\mathbb{Z}_{p^n}$.*

References

1. Barthelemy, L., Eyrolles, N., Renault, G., Roblin, R.: Binary permutation polynomial inversion and application to obfuscation techniques. In: Proceedings of the 2nd International Workshop on Software PROtection. ACM, Vienna, Austria (October 2016)
2. Biondi, F., Josse, S., Legay, A., Sirvent, T.: Effectiveness of synthesis in concolic deobfuscation. Computers & Security **70**, 500–515 (2017)
3. Hardy, G.H., Wright, E.M.: An introduction to the theory of numbers. Oxford, at the Clarendon Press (1960), 4rd ed
4. Markovski, S., Šunić, Z., Gligoroski, D.: Polynomial functions on the units of Z_{2^n}. Quasigroups Related Systems **18**(1), 59–82 (2010)
5. Mullen, G., Stevens, H.: Polynomial functions (mod m). Acta Math. Hungar. **44**(3-4), 237–241 (1984). https://doi.org/10.1007/BF01950276
6. Oruç, H., Phillips, G.M.: Explicit factorization of the Vandermonde matrix. Linear Algebra Appl. **315**(1-3), 113–123 (2000). https://doi.org/10.1016/S0024-3795(00)00124-5
7. Rivest, R.L.: Permutation polynomials modulo 2^w. Finite Fields Appl. **7**(2), 287–292 (2001). https://doi.org/10.1006/ffta.2000.0282
8. Zhou, Y., Main, A., Gu, Y.X., Johnson, H.: Information hiding in software with mixed boolean-arithmetic transforms. In: Information Security Applications, 8th International Workshop, WISA 2007, Jeju Island, Korea, August 27-29, 2007, Revised Selected Papers. pp. 61–75 (2007)